Derna Academy Journal for Applied Sciences
E-ISSN: 3006-3159
DAJAS

A New Conceptual Model for Assessing Risk Factors in Software Development Projects

Mohamed M. Elzahaf [1]

[1] College of Medical Technology, Department of Computer Science, Derna, Libya

*Corresponding author: E-mail addresses: MOHAMEDELZAHAF@YAHOO.COM

Volume: 2 **Issue: 1** **Page Number: 36 - 47**

Keywords:
Conceptual Model, DIT, TAM, Aisk Factors, Risk Assessment, Software Development Projects, FMEA

Received : 09\12\2023
Accepted: 01\01\2024
Published: 25\01\2024

https://doi.org/........../.........vxix.xxxx

ABSTRACT

This paper presents a new conceptual model developed to present a clearer structure and methodology for assessing risk factors in software development projects using the FMEA method. The model was formulated using seven constructs extracted from the combined models of DIT and TAT/TAM that relate to the study. The conceptual model was illustrated in a diagram to describe the constructs and establish how the interacting constructs connect to the study. Two research questions were raised to guide the study, while the descriptions of the constructs and their relationship in the model with the risk assessment activity were used to answer the research questions. The study presents a novel risk assessment methodological contribution to the field of software engineering research. The developed model should be useful to risk managers and their teams who are searching for which appropriate risk assessment tool to use and how best to apply the tool for the conduct of software project risk

1. INTRODUCTION

Even when significant financial, human, and technological resources were invested, many software development initiatives ultimately failed. Software development project failure rates remain extremely high globally, according to a recent statistical analysis of global studies in project development (Giuseppe, 2017; Lehtinen, et al., 2014 de Wet and Visser, 2013). Unfortunately, despite the industry's fast growth over the past three decades, the incidence of software project failure is increasing. Due to this frequent occurrence, the disciplines of software engineering and software process improvement have been established (Lehtinen, et al, 2014). Throughout their careers, the majority of project administrators and leaders have seen one software project fail, and persistent resistance to this approach is a developing issue (KPMG 2011; Geneca, 2011; Cerpa and Verner, 2009).

Scholars involved in software development initiatives (such as Vahidnia et al., 2016; Lehtinen et al., 2014; Lazaros and Prodromos 2011, among others). possess unequivocally demonstrated that proficient handling of significant risk elements in software development projects can reduce the probability of project collapse and elevate the testing methodology by detecting all possible issues arising from high-severity engineering defects throughout the entire Software Development Life Cycle (SDLC). Risk management is a focus topic of emergent study in software development projects for these and other significant reasons. Nevertheless, despite the progress made possible by the availability of sophisticated methods and trustworthy models for software project risk management, data from Research studies from throughout the world indicate that software projects are still failing at a startling rate (Giuseppe, 2017). Software development experts claim that one area of neglect contributing to the persistence of software project failures is the lack of research into suitable mechanisms to be used in order to appropriately identify, estimate, and manage the significant risk factors associated with particular software development models (Keil, et al., 1998). Furthermore, there is insufficient data available to CBSE project managers regarding the viability and applicability of safety and reliability engineering tools (like FMEA) for risk management in the CBSE model because research on their integration in software development projects has not been empirically justified in CBSE models. The purpose of this study was to solve some of the software neglect issues. To address some of the areas of software engineering neglect that experts identified as contributing to the failure of software projects over time, this study was undertaken. Our earlier research on "Adherence Analysis of FMEA with Standard Risk Management Models" is being followed up with this study. The purpose of the empirical research, which was carried out to ascertain whether the FMEA technique could be applied to the risk management process in the CBSE, is presented and analysed in this paper. By shifting managers' attention to other dependable methods for carrying out risk management in CBSE, the study aims to improve the industrial practice of risk management of software projects.

Conceptual Issues

Risk management

Risk management is essentially the control and mitigation of risks (Mcmanus, 2004). It is an ongoing, structured process that starts with risk identification and plan formulation to reduce or manage the consequences of the risk (Lazaros, and Prodromos, 2011). Among the many advantages of implementing an efficient risk management practice in software development projects are helping software development practitioners concentrate on troublesome areas, examining the possible causes of the problems, speculating about possible outcomes to likely solutions, and improving the team's collective perception (Iversen et al. 2004). Figure 1 shows a diagrammatic description of the two-way sub-process of the risk management process (Boehm, 1991).

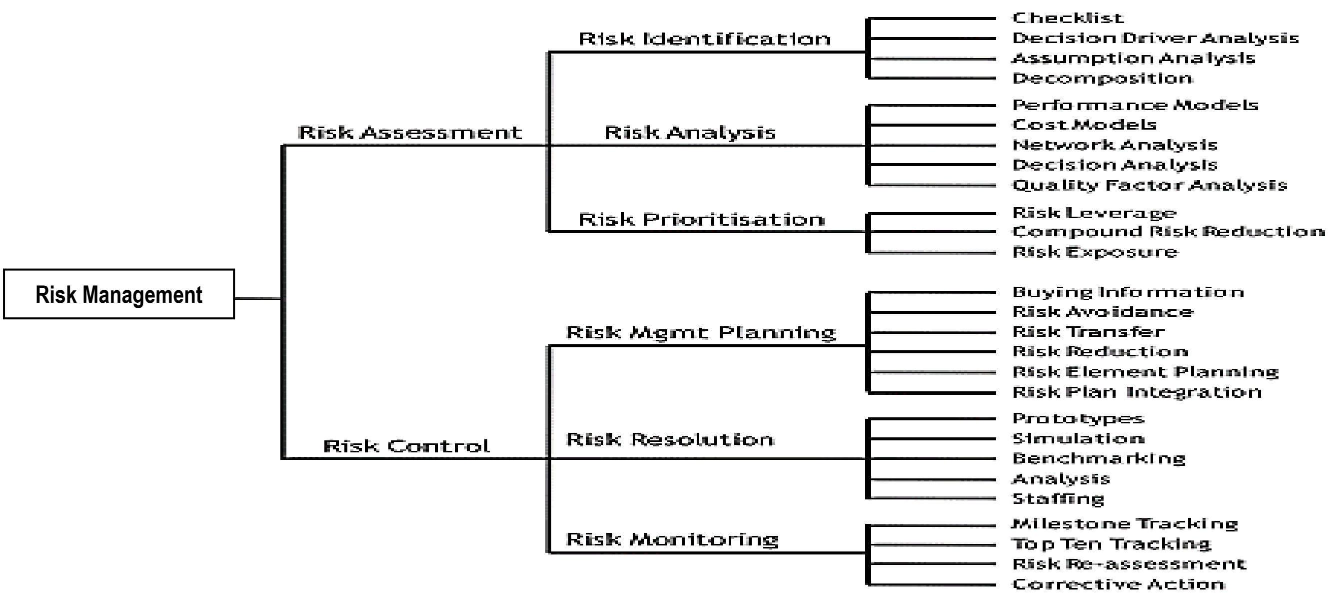


Figure 1: Risk Management Model (Boehm, 1991)

Component Based Software Engineering (CBSE)

The method known as CBSE involves creating intricate software programmes by piecing together reusable parts from several sources into a clearly defined architecture (Gulia & Palakthe, 2017). Due to the growing need for sophisticated and modern software, the CBSE model of software development has become more and more popular in recent years. It has made developing complicated software faster, cheaper, and more modularly possible while reducing delivery times. Making the most of the investment made in reusable components is possible when designs or code are actively reused (Gulia & Palakthe, 2017). From the old waterfall model to highly manageable component-oriented software, the software development process has come a long way. The capacity to reuse earlier efforts made to construct components is a key component of CBSE's progress. Every element is a representation of a A set of services that can be constructed with other components is represented by each component. The software as a whole is constructed from a collection of these interactive components, as seen in Figure 2. Later on, we can change, add, or remove components to suit our needs. This results in reliable software products that are delivered more quickly and at a lower cost, helping to lessen software crises.

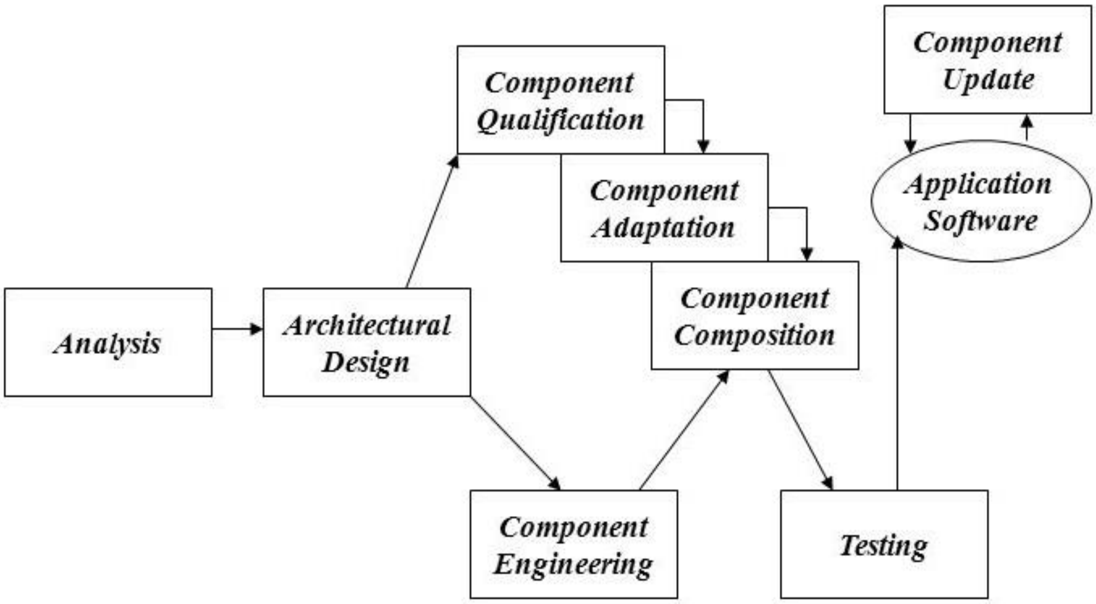


Figure 2: Component Based Development Process

Failure Mode and Effects Analysis (FMEA)

Failure Mode and Effects Analysis (FMEA) is a technique used to investigate possible failures in goods or processes. It was first created for systems engineering (Lutz and Nikora, 2007). The goal of FMEA is to locate and rank potential flaws in processes and products (PUENTE et al., 2001). More specifically, according to the USA Department of Defence (1980) Military Standard, FMEA is "the set of procedures (see Figure 3) by which each potential failure mode in a system is analysed to determine the results or effects thereof on the system and to classify each potential failure mode according to its severity."

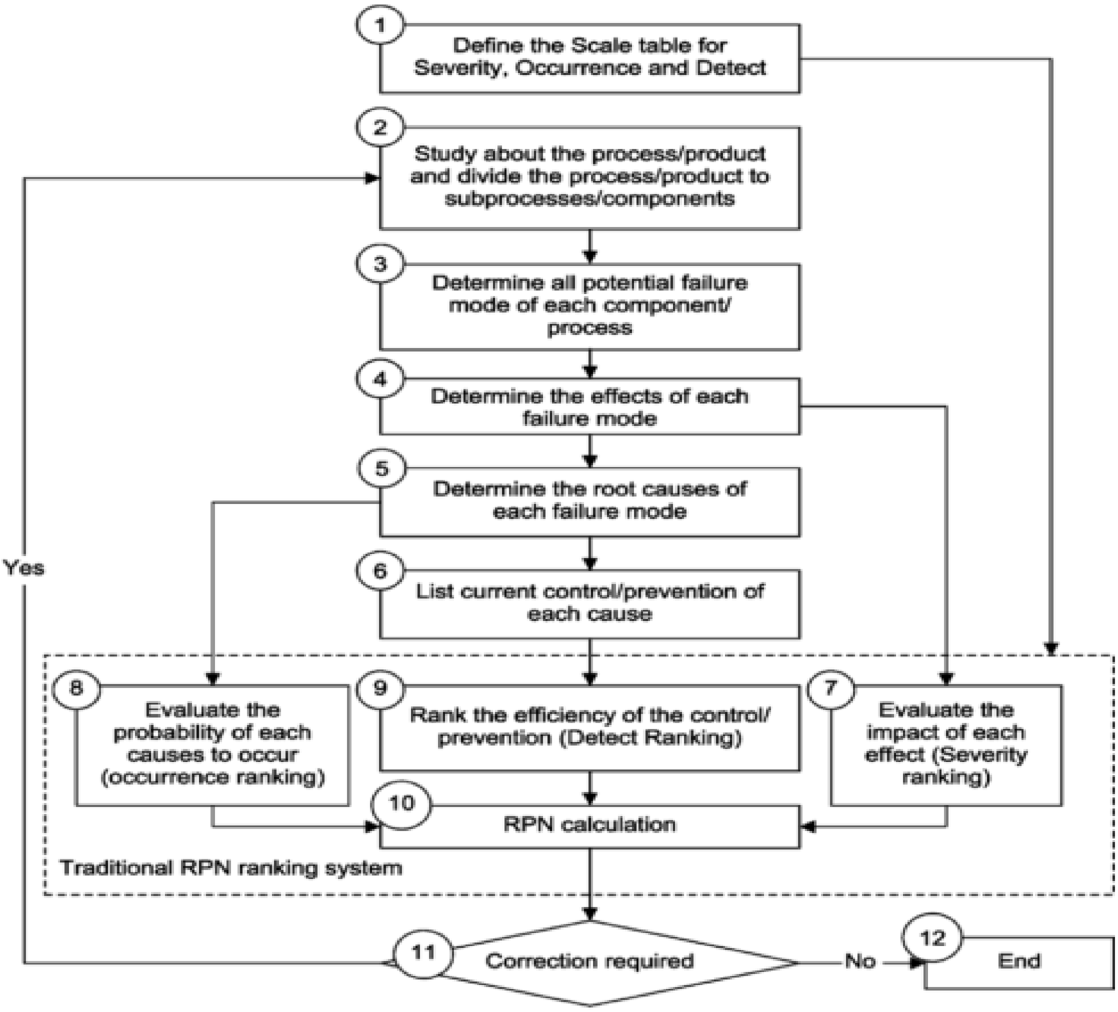


Figure 3: The FMEA basic procedural concept

Problem Statement

The research problem was informed by the challenges confronting risk managers and their teams who are searching which appropriate risk assessment tools and a professional approach can be followed in using the tool for the conduct of software project risk management. Thus, the purpose of this study was to present a conceptual model that describes important constructs and their presumed relationship that will inform risk managers and their teams on how best to use the FMEA to assess the risk factors associated with potential failure modes in the process of software development.

Research Questions

The study's two main research questions were:

• How do the model's constructs support the decision to use the FMEA as a suitable technique for conducting risk assessments for software projects?

• In what ways do the model's constructs give software project risk assessment a more defined goal and structure?

Fundamental presumptions for the research

The study's presumptions were that the conceptual model gave the risk assessment a more defined structure and that the study's focus was on evaluating risk variables linked to possible failure modes in the software development process using the FMEA approach.

Limitations for the study

The conceptual model was developed for assessment of risk factors associated with potential failure modes in the process of software development using the FMEA method. Other risk factors' dimensions were not considered.

Hypotheses

This section discusses the theoretical frameworks used as a lens for the application of FMEA to assess the risk factors associated with potential failure modes in the software development process. Thus, two theories are combined to provide the theoretical foundation for the risk assessment process, which are: Diffusion of Innovation Theory (DIT) and Technology Acceptance Theory/Model (TAT/TAM).

Diffusion of innovation theory (DIT)

The DIT was propounded by Rogers in 1962 to describe how an idea diffuses across a population. The DIT was developed to explain how innovation spreads. The spreading of innovation is a process by which a new idea, product or behaviour is constructed across elements of the target population (Rogers 2003, 1995). According to the theory of Rogers, there are four elements involved in the process of disseminating new idea, practice, or object: (a) it should be classified as innovation; (b) It must be communicated through certain channels; (c) it must be adopted among members within a social system; and (d) it must take into account duration of the time factor. The first and second elements, innovation and communication channels, relate most closely to the risk assessment process. In identifying the channels for communicating innovation, Rogers (1995, 2003) suggested a five communication channel of innovation that is made up of a linear five-stage mental process. Rogers views this linear stage as a mental process from which an innovation can be communicated within a specific period and a social system. In Rogers's view (1995, 2003), the process starts from the first innovation recognition to the formation of an attitude towards it, then to the decision to be adopted or rejected, and later on to the implementation and use of the idea or the new practice, and finally to decision confirmation. Figure 1.0 presents the communication channels for the diffusion of innovation as identified by Rogers (2003).

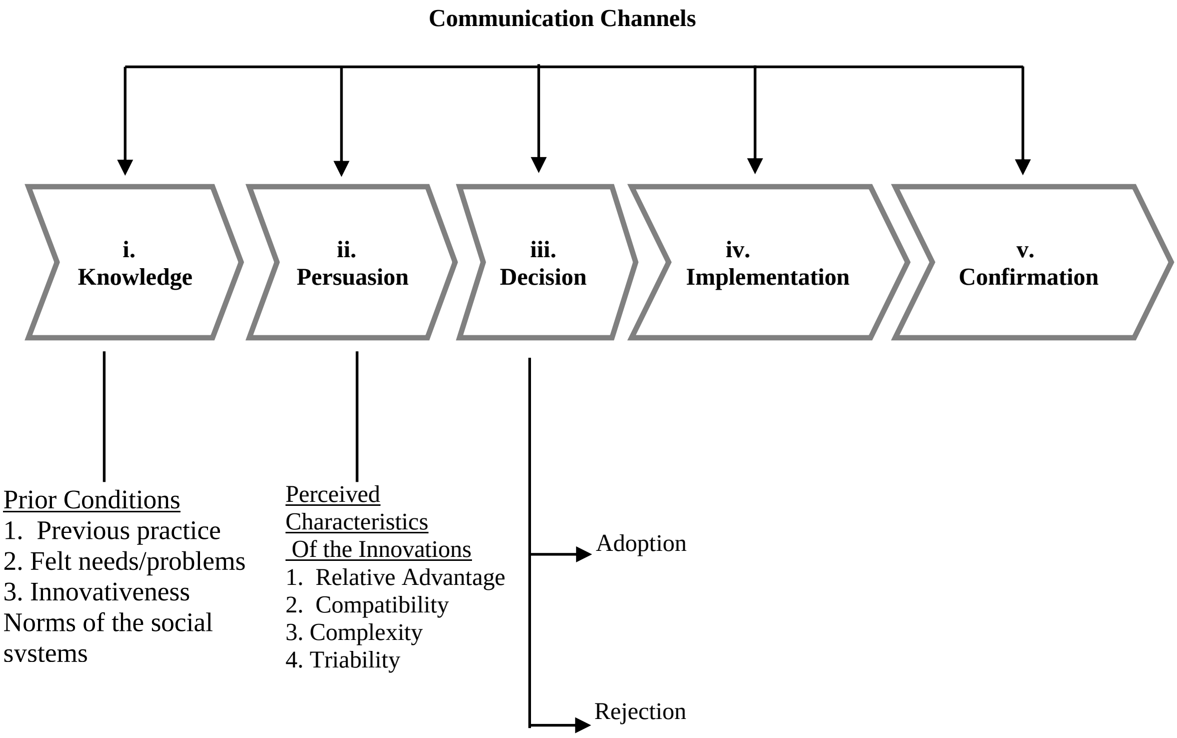


Figure 1.0: Communication channels for diffusion of innovation (Rogers, 1995, 2003)

Rogers (2003) identifies knowledge, persuasion, decision, implementation and confirmation as the fundamental channels through which an innovation, an idea or a product can be communicated and spread or gain momentum over time, through a specific population or community. The DIT theory was considered suitable for adoption for the conduct of risk assessment to (i) Introduce and spread the idea of using FMEA for software project risk management; (ii) Brainstorm on the relative advantages and compatibility of the FMEA with other existing project risk management tools; and (iii) Implement the FMEA to assess the risk factors associated with the potential failure modes in the software development process based on the software development practitioners' perception.

Table 1.0: Rogers' diffusion of theory elements and study examples

Rogers' elements	Study example
The Innovation	Use of FMEA to assess the risk factors associated with potential failure modes in the software development process.
Unit with knowledge and/or experience	selected software development practitioners
Unit with knowledge and/or experience	selected undergraduate SE students
Communication channels	FMEA method workshops, exploratory case study focus group discussion and FMEA method Interactive study

The next sub section discusses the TAT/TAM and its implication on using the FMEA as instrument to assess the risk factors associated with potential failure modes in the software development process.

Technology acceptance theory/model (TAT/TAM)

The TAT/TAM is an IS theory that frames how users accept and use an information system or technology. It is a model extended from the Theory of Reason Action (TRA) done by Davis (1989). TAM came into existence after the introduction of information systems into organizations (Momani & Jamous 2017). The TAM states that the perceptions of users about the usefulness and ease of use of technology are significant determinants of technology acceptance or adoption (Davis 1989). The TAM is applied to investigate and determine the role of the end-user when new technology is initiated. It also analyses the impact of external variables on the acceptance and usage of technology (Momani & Jamous 2017).

Figure 2.0 presents the conceptual model of TAT/TAM.

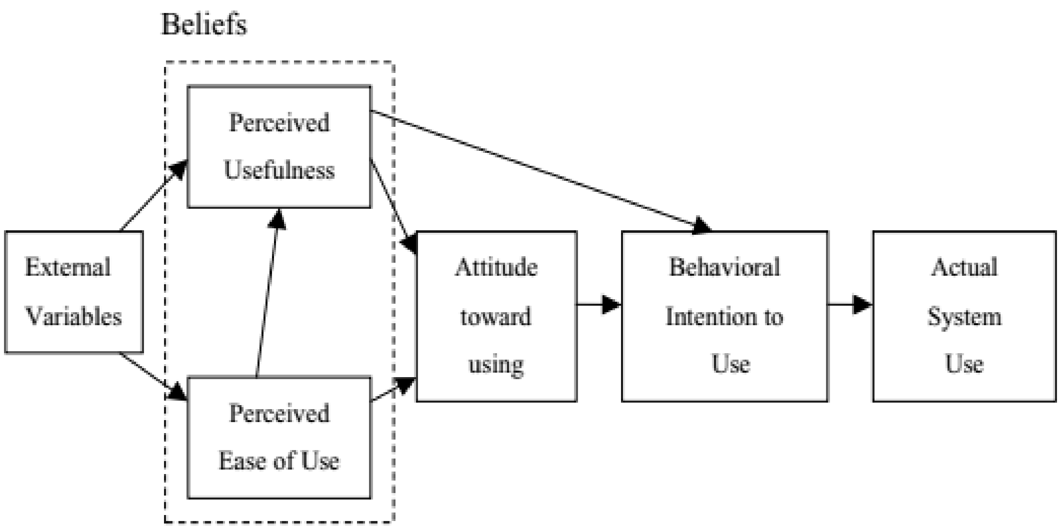


Figure 2.0: conceptual model of TAM (source: Davis 1989)

The following are the various constructs that comprise the TAM, according to Davis (1989): Belief: The person's subjective probability of the outcome of engaging in the desired activity.
Perceived ease of use is the degree to which a user believes that employing a particular technology will be simple (Davis 1998, p 320).
Perceived utility: "The degree to which a person believes that putting a particular system to use would enhance their productivity at work" (Davis 1998, p 320).
External variables are crucial components that are used to evaluate attitude. The availability of TAT/TAM influences people's opinions about it and their intent to utilise it. Individual differences, however, could lead to differences in their attitudes depending on age and gender, among other factors (Davis 1998). When users interact with the technology, the terminal is the real system in use. One element influencing people's adoption of technology is their behavioural intentions. People's overall perceptions of technology are referred to as their attitude (A), and this perception affects their intention to behave (Davis 1989).
The implementation of the modified TAM as applied in this study is presented in the Table 2.0.

Table 2.0: Davies' TAT/TAM constructs with the study examples

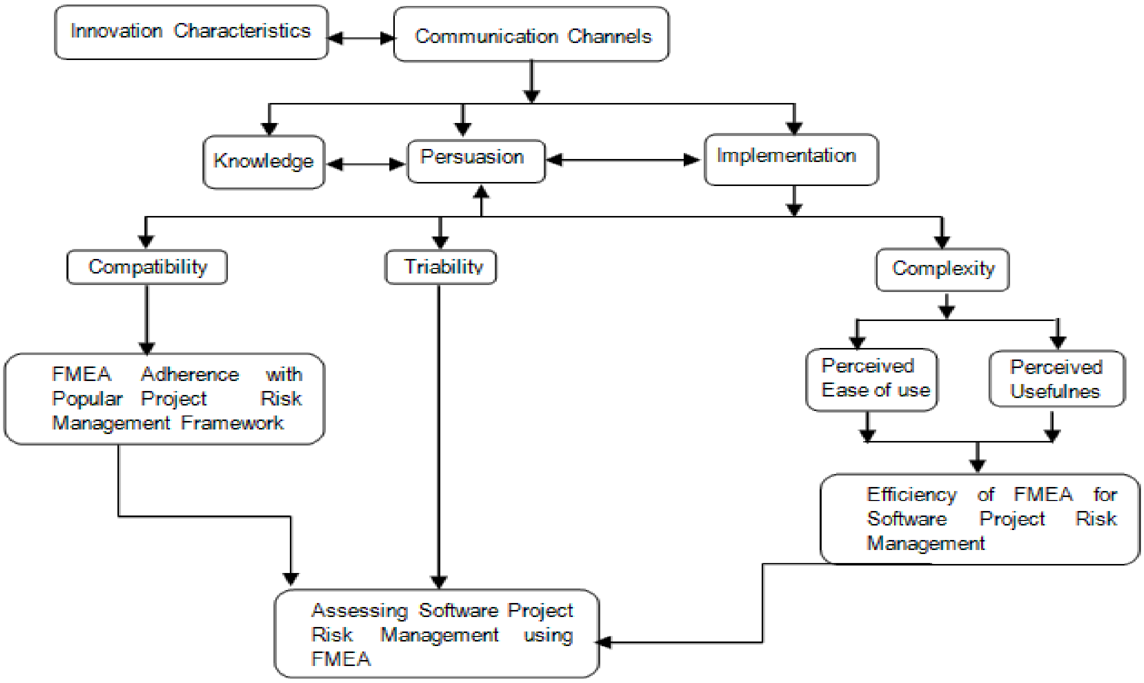
TAT/TAM constructs	Study example
External variables	Introduction and training of the selected Undergraduate SE students
Actual system use	Use of FMEA as a tool to assess the risk factors associated with potential failure modes in the software development process by the selected undergraduate SE students
Perceived ease of use and perceived usefulness	Perceptions of the trained undergraduate SE students on the efficiency of FMEA as a tool for the conduct of software project risk management

2. METHOD

This section discusses the method adopted in developing a new conceptual model for assessing the risk factors in software development projects and discusses how the constructs therein address the research questions. As stated earlier, the conduct of the risk assessment is specifically concerned with the seven core constructs extracted from both the DIT and TAT/TAM. Five constructs extracted from the DIT are knowledge, persuasion, triability and implementation. The two constructs extracted from TAM are perceived ease of use and perceived usefulness. The five constructs from DIT are direct determinants of users' rate of adoption of innovation (Rogers, 2003). Similarly, the constructs: perceived ease of use and perceived usefulness are direct determinants of users' behavioral use of Information systems or technology. In this study, a conceptual framework was formulated through the convergence of seven constructs extracted from the combined models DIT and TAT/TAM (five from DIT and two from TAM/TAT) that relate to the study. These constructs are relevant to the study because they are direct determinants of users' perception of new ideas, approaches or innovations (Rogers 2003) and actual behavioral use of a system (Davies 1989). In the case of this study, the five fundamental constructs extracted from DIT are considered determinants to assess the perceptions that software development practitioners hold towards the use of FMEA to assess software project risk factors and the effect they considered the risk factors will have on project outcome, while the other two constructs adopted from TAM are considered direct determinants of behavioural usage of the FMEA for the conduct of software project risk management.

3. RESULT

To shed light on the organization of the risk assessment procedure, a diagram illustrating the convergence of the seven elements that comprised the conceptual model was created. The innovation features, communication channels, knowledge, persuasion, and implementation, as well as perceived ease of use and perceived utility, are the seven constructs that directly impact the study. The conceptual model that was created for evaluating the risk variables connected to probable failure modes in the software development process is depicted in Figure 3.0.



A conceptual model for evaluating the risk variables connected to probable failure modes during the software development process is shown in Figure 3.0. The model's interactive components were utilized to address the following research issues, and the diagrammatic representation explains how the constructs in the model relate to the risk assessment activity:

• **Question 1**: How do the model's constructs support the decision to use the FMEA as a useful method for conducting risk assessments for software projects?

The structure: According to Rogers (2003), compatibility with current values and practices refers to how well a novel concept or innovation aligns with past experiences, current values, and the requirements of potential users. An idea is more likely to be adopted quickly if it aligns with the norms and values than if it doesn't (Rogers 1995, 2003). To tackle Research Question 1, the model's construct compatibility pertains to a survey conducted among software practitioners regarding their opinions of FMEA compliance with formal project risk management models (Lawal 2022). The compatibility study involving experienced software practitioners to investigate adherence to the FMEA procedural requirements with formal project risk management frameworks to justify the choice of the FMEA as an effective tool for the conduct of software project risk assessment, thus providing an answer to RQ1.

• **RQ2:** How do the constructs in the model provide a clearer structure and mission for the conduct of software project risk assessment?

Each of the seven constructs presented in the conceptual framework is fundamental to the conduct of the risk assessment process and can be used to address the RQ2. Therefore, the following is the description of how the constructs in the conceptual model provide a clearer structure for the conduct of software project risk assessment. Characteristics of innovation: The characteristics of innovation are defined by Rogers (1998) as reasons for the adoption of innovation at all stages. Rogers identified the following characteristics set for any innovation to be adopted: Relative advantage, Compatibility, Triability, Complexity and observability as discussed by Rogers (2003).

In this study, the characteristics of FMEA reviewed from the literature are regarded as the perceived characteristics of FMEA.

Communication Channel: This is regarded as a means through which the innovation gets communicated to people to be adopted (Rogers 1998, 2003). Rogers identified communication channels as an important element of the diffusion process. Rogers (1998) identified a linear five-stage mental process through which the perceived innovation characteristics can be spread among members of a social system.

In this study, the perceived FMEA characteristics are communicated to the participating software practitioners using workshops, exploratory case studies, focus group discussions and FMEA interactive studies with student developers.

Knowledge: This is the first stage of spreading the innovation. Here, members of the social systems are exposed to new ideas/approaches that are considered as innovation. At the initial stage, it was assumed that members had very limited information to decide to adopt an innovation and were also not sufficiently inspired to find out more. In this stage, members of the community are exposed to new ideas or products and gather more information about them (Rogers 1998).

Persuasion: Persuasion is the location at which the members of the community are open to a new approach or idea (Rogers 1998). This stage involves the formation of an attitude towards innovation itself. It is the stage where the users actively find out more knowledge that influences their decision on the innovation. In this study, lecture presentations and training that informed student developers on the approach and usage of FMEA for the conduct of software project risk management were delivered through the various channels of diffusion stated above. These activities are considered effective enough to expose and train the participants on how to use the FMEA for the conduct of software project risk management.

Implementation: In this stage, The innovator uses the invention and assesses its advantages (Rogers 1998). They might also gather more data to support their position or refute the innovation's applicability. Three innovative features are applied to the FMEA for the management of software project risk in the context of this study. First, compatibility: software development professionals participated in a survey using a questionnaire to gauge how closely the FMEA framework adhered to the widely used project risk management frameworks (Lawal 2022). Second: Triability: in this case, software professionals evaluate software project risk factors using evidence-based research that makes use of the FMEA.

The perspectives of practitioners regarding the impact of software project risk variables on project outcomes were then evaluated by a questionnaire survey (Lawal 2022). Finally, undergraduate SE students were instructed to apply FMEA methodologies for their practical SE course following their instruction in an FMEA interactive study (Lawal 2002). They have to manage the risk of the software project using FMEA. Following the practical engagement, the involved students were asked to respond to a questionnaire expressing their opinions regarding the FMEA tool's perceived utility and ease of use for carrying out the risk assessment project. Judged to be user-friendly: "The extent to which an individual thinks that utilising a specific system would require no effort" (Davis 1998, p 320).

Perceived usefulness: "The extent to which an individual feels that utilising a specific system would improve their performance at work" (Davis 1998, p 320). The effectiveness of the FMEA for managing software project risk is determined by student developers' assessments of its perceived utility and ease of use.

Hypotheses

Ho1: The mean scores of male and female practitioners in the CBSE on the practicality of conducting risk management exercises using the FMEA approach are not significantly different from one another.

The t-calculated value is 0.68 and the t-critical value is 1.96, as Table 3 demonstrates. Since 0.68 is smaller than 1.96 at the 0.05 threshold of significance, the null hypothesis is not rejected. This indicates that there is little difference in the mean scores of male and female practitioners when it comes to the application of the FMEA technique for risk management in CBSE.

Table 3: T-test Mean Response of Male and Female Practitioners in CBSE on the feasibility of applying FMEA methodology for the conduct of risk management exercise in the CBSE

Respondents	**N**	**X**	**SD**	**d/f**	**t-cal**	**t-crit**	**Decision**
Male Practitioner	39	3.54	0.77	488	0.68	1.96	Ns
Female Practitioner	11	3.47	0.82				

Ho2: When it comes to how well the use of FMEA technique in the CBSE can reduce the likelihood of project failure in the CBSE, there is no discernible difference in the mean scores of male and female practitioners.

The t-calculated value in Table 4 is 0.45, whereas the t-critical value is 1.96. This demonstrates that, at the 0.05 level of significance, the t-calculated (0.45) is smaller than the t-critical (1.96). As a result, the null hypothesis is not rejected, suggesting that gender has no discernible impact on how effectively the FMEA methodology applied in the CBSE may reduce the likelihood of project failure in CBSE models.

Table 4: T-test mean Rating of Male and Female Practitioners in CBSE on the extent to which FMEA methodology can efficiently minimize the incidence of project failure in CBSE

Respondents	**N**	**X**	**SD**	**d/f**	**t-cal**	**t-crit**	**Decision**
Male Practitioner	39	3.43	0.71	488	0.45	1.96	Ns
Female Practitioner	11	3.38	0.76				

4. DISCUSSION

The seven elements related to risk factor assessment in software development projects that were taken from the combined model's DIT and TAT/TAM (two from TAM/TAT and five from DIT) were used to create the new conceptual model. To provide a conceptual framework that explains the suitable tool and the competent and reliable risk management technique used to conduct risk assessment in software development projects, the convergence constructs were adjusted. The TAM was included to fully examine the extent to which TAM can help understand the usage behaviour of trained undergraduate software engineering students, while the DIT was adopted due to its ability to explain the FMEA spread process regardless of discipline, culture, or national border. Pupils who used the FMEA as a Students who assessed and managed risk in software development projects using the FMEA. The efficacy of the FMEA for the conduct of software project risk management is believed to be directly influenced by the other two constructs, which were extracted from TAM. Software development practitioners' opinions of the application of FMEA to analyse software project risk factors and the consequences they believe the risk factors will have on project success are hypothesised to be determined by five key constructs extracted from the DIT.

The interactive relations of the seven constructs presented in the conceptual framework were used address the research questions. For example, compatibility is concerned with software practitioners' perceptions of FMEA adherence to formal project risk management frameworks (Lawal 2022). Thus, determining the appropriateness of the FMEA for the conduct of risk assessment. The construct triability identifies and assesses the software project risk factors' effect on project outcome as narrated and percieved by the software participants (Lawal 2022). The perceived ease of use and usefulness in the conceptual model is concerned with the efficiency of the FMEA for the conduct of software project risk management as percieved by the trained undergraduate SE students (Lawal 2022). Interestingly, all the constructs function harmoniously in an interrelated and interdependent way towards usiing FMEA to assessing the risk factors associated with potential failure modes in the software development process

Recommendation

1. Adopting the new conceptual model for the conduct of risk management process

Risk managers and their team members who are searching for which appropriate tool to use and the best approach that can be followed in using the tool for the conduct of software project risk management are encouraged to use the developed conceptual model for their project risk management

2. Modelling professional risk control technique

This study provided a new conceptual model for conducting risk assessment in the process of software development using the FMEA method Future research should focus on risk modelling professional risk control techniques for the risky potential failure modes in all the stages of the software development process.

5. Conclusion

Based on the findings of the software project risk assessment that was conducted using the developed conceptual model (Lawal 2022), it was concluded that.

1- The new conceptual model provided a professional and stable risk management approach to risk managers and their teams who are searching for appropriate risk assessment tools and how best to apply the tool for the conduct of software project risk management

2- Another significant conclusion was that the developed model presents a novel risk assessment methodological contribution in the field of SE research.

6. REFERENCES

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS quarterly, 319–340.

Davis, F. D. (1985). A technology acceptance model for empirically testing new end- user information systems: Theory and results (Doctoral dissertation, Massachusetts Institute of Technology).

Castsoftware, (2015). Report on Software Quality and Developer productivity. Available at: www.castsoftware.com/blog/software-quality-and-developer-productivity-together-improve-efficiency

CAST (2015). Software Quality and Developer Productivity: Together Improve Efficiency. Software Intelligence Pulse. Retrieved October, 2016 from www.castsoftware.com/blog/software-quality-and-developer-productivity-together-improve-efficiency

Kontio, J., Getto, G & Landes, D. (1998). Experiences in Improving Risk Management Processes Using the Concepts of the Riskit Method. Proceedings of the ACM SIGSOFT 6th International Symposium on Foundations of Software Engineering, ACM Press, New York, November 1998, 163-174.

Lawal, B. (2022). Assessing the Risk Factors Associated with Potential Failure Modes in the Software Development Process. Unpublished Ph.D. Thesis, School of Computing, University of South Africa, Pretoria, South Africa

Lazaros, S. & Prodromos, D. P. (2011). Software Development Project Risk Management: A New Conceptual Framework. Journal of Software Engineering and Applications, 2011(4), 293-305 doi:10.4236/jsea.2011.45032

Miles, M. B & Huberman, A. M. (1994). Qualitative data analysis: An expanded source book (2nd ed.). Thousand Oaks, California: Sage Publications

Project Management Institute PMI.(2017). A Guide to the Project Management Body of Knowledge (PMBOK® Guide) (sixth edition.). Project Management Institute, EUA

Rogers, E. M. (2003). Diffusion of innovations (5th ed.). Free Press

Rogers, E. M. (1995). Diffusion of Innovations: Modifications of a model for telecommunications. Die Diffusion von Innovationen in der Telekommunikation, 17, 25–38